

Claims:

1 1. System for measuring the quality of a data
2 network (3), the system comprising a client
3 computer (1) and a server computer (2), the client
4 computer (1) and the server computer (2) being arranged
5 to communicate via the data network (3) using a
6 connection-less transmission protocol,

7 the client computer (1) being arranged to send one
8 or more data packets (1001) to the server computer (2),
9 the data packets comprising a first timestamp (103)
10 indicating the time of sending,

11 the server computer (2) being arranged to receive
12 at least one of the data packets, the server computer
13 further being arranged to send, for each received data
14 packet, a modified data packet (1002) to the client
15 computer (1), the modified data packet comprising the
16 first timestamp (103) and a second timestamp (105)
17 indicating the time of receiving,
18 wherein

19 the server computer (2) is arranged to send one or
20 more timeout data packets (1003) to the client computer
21 when a predefined amount of time lapsed (1004) since
22 receiving the last data packet in the server
23 computer (2).

1 2. System according to claim 1, wherein the server
2 computer (2) comprises a server processor (2a)
3 connected to a server memory (2b), the server
4 processor (2a) being arranged to read from the server
5 memory (2b) a timeout value indicating the predefined

6 amount of time, a server interval value and a quantity
7 value,

8 the server computer (2) being arranged to send the
9 timeout data packets (1003) with a server interval time
10 equal to the server interval value, until a next data
11 packet (1001) is received, and

12 the maximum number of timeout data packets equals
13 the quantity value.

1 3. System according to claim 2, wherein the client
2 computer (1) comprises a client processor (1a)
3 connected to a client memory (1b), the client
4 processor (1a) being arranged to read from the client
5 memory (1b) for each data packet a client interval
6 value, a client packet size value and a packet type
7 value,

8 the client computer (1) being arranged to send the
9 data packets (1001) with a client interval time equal
10 to the client interval value,

11 the data packets comprising the packet type
12 value (101) and a client payload (104), the client
13 payload (104) filling the data packets up to a first
14 predefined size equal to the client packet size value.

1 4. System according to claim 3, wherein the packet
2 type value is a representation of "normal packets" or
3 "burst packets".

1 5. System according to claim 3, wherein the client
2 memory (1b) is a file stored on a disk.

1 6. System according to claim 1, wherein each data
2 packet comprise a sequence number (102).

1 7. System according to claim 1, wherein the server
2 processor (2a) is arranged to read from the server
3 memory (2b) a server packet size value,
4 the modified data packet comprising a server
5 payload (106), the server payload filling the modified
6 data packet up to a second predefined size equal to the
7 server packet size value.

1 8. System according to claim 3, wherein the client
2 payload (104) and/or the server payload (106) comprises
3 a predefined bit pattern.

1 9. System according to claim 3, wherein the client
2 payload (104) and/or the server payload (106) comprises
3 a random bit pattern.

1 10. System according to claim 1, wherein the
2 connection-less transmission protocol is UDP/IP.

1 11. System according to claim 2, wherein the client
2 computer (1) is arranged to send a configuration packet
3 (1005) to the server computer (2), the configuration
4 packet comprising the timeout value (201), the server
5 interval value (202), the quantity value (203) and/or
6 the server packet size value (204),
7 the server computer (2) is arranged to receive the
8 configuration packet,

9 and the server processor (2a) is arranged to write
10 to the server memory (2b) the timeout value, the server
11 interval value, the quantity value and/or the server
12 packet size value.

1 12. System according to claim 2, wherein, for the
2 purpose of configuring the server computer (2), the
3 client computer (1) is arranged to communicate with the
4 server computer (2) via the data network (3) using a
5 connection-oriented transmission protocol,

6 the server computer (2) is arranged to receive
7 configuration data from the client computer (1), the
8 configuration data comprising the timeout value, the
9 server interval value, the quantity value and/or the
10 server packet size value,

11 and the server processor (2a) is arranged to write
12 to the server memory (2b) the timeout value, the server
13 interval value, the quantity value and/or the server
14 packet size value.

1 13. System according to claim 12, wherein the
2 connection-oriented transmission protocol is TCP/IP.

1 14. System according to claim 1, wherein the client
2 computer (1) is arranged to store a logfile, the
3 logfile comprising all modified data packets received
4 from the server computer (2), the logfile further
5 comprising all timeout data packets received from the
6 server computer (2).

1 15. System according to claim 14, wherein all modified
2 data packets received from the server computer (2) are
3 stored in a first logfile, and all timeout data packets
4 received from the server computer (2) are stored in a
5 second logfile.

1 16. System according to claim 15, wherein the client
2 computer (1) stores each data packet in a ring buffer,
3 the client computer (1) being arranged to check
4 each modified data packet received from the server with
5 the contents of the ring buffer,
6 the client computer (1) being arranged to delete
7 the data packet from the ring buffer if the
8 corresponding modified data packet is not received
9 within the length of the ring buffer, and in this case
10 store the data packet in the second logfile.

1 17. System according to claim 16, wherein the client
2 computer (1) is arranged to calculate a round-trip-time
3 by subtracting the first timestamp from a timestamp
4 indicating when the modified data packet is received in
5 the client computer (1), and the client computer (1) is
6 arranged to store the round-trip-time in the first
7 logfile.

1 18. System according to claim 16, wherein the client
2 computer (1) is arranged to calculate a
3 client-to-server-latency by subtracting the second
4 timestamp from the first timestamp, and the client
5 computer (1) is arranged to store the
6 client-to-server-latency in the first logfile.

1 19. System according to claim 18, wherein the client
2 computer (1) is arranged to read from the first logfile
3 the packet type value, the client-to-server-latency and
4 the round-trip-time,

5 and the client computer (1) is arranged to
6 calculate for each modified data packet with the packet
7 type value equal to the representation of "normal
8 packet" a server-to-client-latency by subtracting the
9 client-to-server-latency from the round-trip-time.

1 20. System according to claim 18, wherein the client
2 computer (1) is arranged to read from the first logfile
3 the packet type value,

4 and the client computer (1) is arranged to
5 calculate a downlink bandwidth using:

6
$$(\text{\#BytesPerPacket} + \text{Overhead}) * (\text{\#BurstPackets}) / (t_2 - t_1)$$

7 where

8 #BytesPerPacket equals the length of the modified
9 data packet in bytes;

10 Overhead equals the length of protocol header
11 overhead in bytes;

12 #BurstPackets equals the number of modified data
13 packets stored in the first logfile with the packet
14 type value equal to the representation of "burst
15 packet";

16 t1 equals a timestamp indicating when the first
17 modified data packet is received in the client
18 computer (1);

19 t2 equals a timestamp indicating when the last
20 modified data packet is received in the client
21 computer (1).

1 21. System according to claim 18, wherein the client
2 computer (1) is arranged to read from the first logfile
3 the packet type value,

4 and the client computer (1) is arranged to
5 calculate an uplink bandwidth using:
6
$$\frac{(\#BytesPerClientPacket + Overhead) * (\#BurstPackets)}{((t2 + t4) - (t1 + t3))}$$

7

8 where

9 $\#BytesPerClientPacket$ equals the length of the
10 data packet in bytes;

11 $Overhead$ equals the length of protocol header
12 overhead in bytes;

13 $\#BurstPackets$ equals the number of modified data
14 packets stored in the first logfile with the packet
15 type value equal to the representation of "burst
16 packet";

17 $t1$ equals a timestamp indicating when the first
18 modified data packet is received in the client
19 computer (1);

20 $t2$ equals a timestamp indicating when the last
21 modified data packet is received in the client
22 computer (1);

23 $t3$ equals the client-to-server-latency of the
24 first modified data packet;

25 $t4$ equals the client-to-server-latency of the last
26 modified data packet.

1 22. System according to claim 18, wherein the client
2 computer (1) is arranged to read from the first logfile
3 the round-trip-time and the sequence number,

4 and the client computer (1) is arranged to
5 calculate a roundtrip outage using:
6 if(Rtt2-Rtt1)>t5 then roundtrip outage=true
7 where
8 Rtt2 equals the round-trip-time;
9 Rtt1 equals the round-trip-time of the previous
10 modified data packet, the client computer being
11 arranged to identify the previous modified data packet
12 by the sequence number;
13 t5 equals a predefined amount of time.

1 23. System according to claim 18, wherein the client
2 computer (1) is arranged to read from the first logfile
3 the client-to-server-latency and the sequence number,
4 and the client computer is arranged to calculate a
5 downlink outage using:
6 if(Rtt4-Rtt3)>t5 then downlink outage=true
7 where
8 Rtt4 equals the client-to-server-latency;
9 Rtt3 equals the client-to-server-latency of the
10 previous modified data packet, the client computer
11 being arranged to identify the previous modified data
12 packet by the sequence number;
13 t5 equals a predefined amount of time.

1 24. System according to claim 19, wherein the client
2 computer (1) is arranged to read from the first logfile
3 the sequence number,
4 and the client computer (1) is arranged to
5 calculate an uplink outage using:
6 if(Rtt6-Rtt5)>t5 then uplink outage=true

7 where

8 Rtt6 equals the server-to-client-latency;

9 Rtt5 equals the server-to-client-latency of the
10 previous modified data packet, the client computer
11 being arranged to identify the previous modified data
12 packet by the sequence number;

13 t5 equals a predefined amount of time.

1 25. System according to claim 22, wherein the client
2 computer (1) is arranged to determine a number of lost
3 packets between two modified data packets, by analyzing
4 the sequence numbers.

1 26. System according to claim 17, wherein the client
2 computer (1) is arranged to combine the calculated
3 values with low level measurement data from a mobile
4 telephone and/or geographic information data.

1 27. Method for measuring the quality of a data
2 network (3) in a system comprising a client
3 computer (1) and a server computer (2), the client
4 computer (1) and the server computer (2) being arranged
5 to communicate via the data network (3) using a
6 connection-less transmission protocol, the method
7 comprising the steps of
8 sending one or more data packets (1001) from the
9 client computer (1) to the server computer (2), the
10 data packets comprising a first timestamp (103)
11 indicating the time of sending,
12 receiving at least one of the data packets in the
13 server computer (2),

14 sending, for each received data packet, a modified
15 data packet (1002) from the server computer (2) to the
16 client computer (1), the modified data packet
17 comprising the first timestamp (103) and a second
18 timestamp (105) indicating the time of receiving,
19 sending one or more timeout data packets (1003)
20 from the server computer (2) to the client computer (1)
21 when a predefined amount of time lapsed (1004) since
22 receiving the last data packet in the server
23 computer (2).

1 28. Method according to claim 27, the method further
2 comprising the steps of
3 reading from a server memory (2b) in the server
4 computer (2) a timeout value indicating the predefined
5 amount of time, a server interval value and a quantity
6 value,
7 sending the timeout data packets (1003) with a
8 server interval time equal to the server interval
9 value, until a next data packet is received, and the
10 maximum number of timeout data packets equals the
11 quantity value.

1 29. Method according to claim 28, the method further
2 comprising the steps of
3 reading from a client memory (1b) in the client
4 computer (1) for each data packet a client interval
5 value, a client packet size value and a packet type
6 value,
7 sending the data packets (1001) with a client
8 interval time equal to the client interval value, the

9 data packets comprising the packet type value (101) and
10 a client payload (104), the client payload (104)
11 filling the data packets up to a first predefined size
12 equal to the client packet size value.